

Pregunta 1: ¿Cual sería la salida del siguiente programa?

```
console.log(f(false));  
function f(input: boolean) {  
    let x= 10;  
  
    if (input) {  
        let y = x+ 1;  
        return y;  
    }  
    return y;  
}
```

Respuestas:

- a) 11
- b) 1
- c) Undefined
- d) Error

Pregunta 2: En Angular, puede pasar datos de un componente padre a un componente hijo usando...

Respuestas:

- a) @Output()
- b) @Input()
- c) getParentData()
- d) @passParentData()

Pregunta 3: Para construir una aplicación en modo de producción, se utiliza el siguiente comando:

Respuestas:

- a) ng build to prod
- b) ng prod build
- c) ng build --prod=true
- d) ng --build -p

Pregunta 4: El siguiente comando se usa para ejecutar el análisis de código estático (Static Code) de la aplicación Angular:

Respuestas:

- a) ng build
- b) ng profile
- c) ng lint
- d) ng static

Pregunta 5: En Angular Routing, la siguiente etiqueta es usada para representar un componente coincidente a través de la ruta activa.

Respuestas:

- a) `<router></router>`
- b) `<router-output></router-output>`
- c) `<router-outlet></router-outlet>`
- d) `<router-display></router-display>`

Pregunta 6: ¿Que atributo es siempre "true" cuando la clase Form Control de ngModel es modificada?

Respuestas:

- a) `.ng-invalid`
- b) `.ng-pending`
- c) `.ng-pristine`
- d) `.ng-dirty`

Pregunta 7: Puede crear una referencia local en HTML hacia una etiqueta HTML usando la variable que empieza con el carácter...

Respuestas:

- a) `@`
- b) `#`
- c) `*`
- d) `$`

Pregunta 8: ¿Cuál es la forma correcta de llamar al método de "RouterModule" que va a proveer todas las rutas en "AppModule"?

Respuestas:

- a) `RouterModule.forRoot`
- b) `RouterModule.forChild`
- c) `RouterModule`
- d) `RouterModule.import`

Pregunta 9: ¿Cuál de las siguientes es una forma correcta de generar el componente mycomp en Angular desde CLI?

Respuestas:

- a) `ng mycomp -gen`
- b) `ng mycomp -gcd`
- c) `ng g c mycomp`
- d) `ng mycomp --comp --deploy`

Pregunta 10: ¿Cómo puedes hacer referencia de un DIV en el código?

```
<div id="container" #container [formGroup]="form">
  <label>Name</label>
  <input type="text" formControlName="name">
</div>
```

Respuestas:

- a) \$('#container')
- b) document.getElementById('container')
- c) @ViewChild('container')
- d) jQuery('#container')

Pregunta 11: La siguiente porción de código indica que hay dos rutas que cargan el mismo componente. Si revisa la consola al ir a /path-a y después a /path-b verá lo que se muestra al final en console output. ¿Cómo puede evitar que al pasar a path-b, siga corriendo la consola con el contador de path-a?

```
// app-routing.module.ts
const routes: Routes = [
  {path: 'path-a', component: HelloComponent},
  {path: 'path-b', component: HelloComponent},
]
// hello.component.ts
@Component({
  selector: 'hello',
  template: `<h1>Hello</h1>`,
})
export class HelloComponent implements OnInit {
  timed$ = timer(1, 1000);
  ngOnInit() {
    this.timed$.subscribe(r => console.log(r))
  }
}
// Console output
0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 0 -> 7 -> 1 -> 8 -> 2 -> 9 (...)
```

Respuestas:

- a) De suscribirse en el evento OnDestroy().
- b) Agregar un parámetro para determinar la suscripción actual.
- c) Buscar el componente y eliminarlo en el DOM.
- d) Ninguna de las anteriores.

Pregunta 12: ¿Cuál de las siguientes opciones nos puede proveer una lista de productos dentro del componente de productos?

```
<products-list [list]="productsList"></products-list>
```

Respuestas:

- a) public list: any[]
- b) @HostProperty() data: any[];
- c) @Output() data: any[];
- d) @Input() data: any[]

Pregunta 13: ¿Qué tipo de “bindings” están presentes en la siguiente plantilla?

```
import { Component } from '@angular/core';
```

```
@Component({
  selector: 'my-app',
  template: `
    <h1 #stuff>Hello {{name}}</h1>
    <input type="text" [(ngModel)]="name">
    <button (click)="reset()">Set Name as React</button>
  `,
})
export class AppComponent {
  name = 'Angular';

  reset = () => this.name = 'React';
}
```

Respuestas:

- a) “Interpolation”, “Two Way binding” y “Event Binding”
- b) “Two-way binding”, “Template Variable” y “Interpolation”
- c) “One way binding”, “Event Binding” y “Two Way binding”
- d) “Event Binding”, “Two way binding” y “interpolation”

Pregunta 14: ¿Cuál sera el output, considerando que isTrue = 1?

```
<h1 *ngIf="isTrue; then userBlock; else guestBlock">All</h1>
<ng-container #userBlock>Bob</ng-container>
<ng-container #guestBlock>Castor</ng-container>
```

Respuestas:

- a) All Castor
- b) Bob
- c) All Bob
- d) Aparecerá un error al usuario.

Pregunta 15: ¿Cuál es la mejor forma de enviar un “Authentication Token” en cada petición HTTP?

Respuestas:

- a) Usando 'rxjs' y pasarlo en cada petición con un 'getter'.
- b) Implementando un 'Interceptor'.**
- c) Enviándolo como argumento en el servicio.
- d) Ninguna de las anteriores.

Pregunta 16: ¿Cuál es la secuencia correcta del ciclo de vida de los 'hooks'?

```
ngOnInit() {console.log(A)}  
ngAfterContentInit() {console.log(B)}  
ngAfterContentChecked() {console.log(C)}  
ngDoCheck() {console.log(D)}  
ngOnChanges() {console.log(E)}  
ngAfterViewInit() {console.log(F)}  
ngAfterViewChecked() {console.log(G)}  
ngOnDestroy() {console.log(H)}
```

Respuestas:

- a) A -> B -> C -> D -> E -> F -> G -> H
- b) E -> A -> D -> B -> C -> F -> G -> H**
- c) E -> A -> D -> F -> G -> B -> C -> H
- d) A -> B -> F -> G -> C -> D -> E -> H

Pregunta 17: Tiene 2 componentes anidados ya cargados. ¿Qué pasará cuando presione el botón “Button Two Change Name”?

```
import { Component, Input } from '@angular/core';
```

```
@Component({  
  selector: 'my-app',  
  template: '<parent name="{{ name }}"></parent>',  
  styleUrls: [ './app.component.css' ]  
})  
export class AppComponent {  
  name = 'Angular';  
}
```

```
@Component({  
  selector: 'parent',  
  template: <h1>Parent Component</h1><h1>Hello {{name}}!</h1> <button  
(click)="changeName()">Button One Change Name</button> <hr> <app-child  
name="{{name}}"></app-child>,  
  styles: [h1 { font-family: Lato; }]  
})
```

```

export class ParentComponent {
  @Input() name: string;
  toggle: boolean = false;

  changeName() {
    this.toggle = !this.toggle;
    this.name = this.toggle ? 'Name A' : 'Name B'
  }
}

@Component({
  selector: 'app-child',
  template: <h>Child Component</h1> <h1>Hello {{name}}!</h1><button
(click)="changeName()">Button Two Change Name</button>,
  styles: [h1 { font-family: Lato; }]
})
export class ChildComponent {
  @Input() name: string;
  toggle: boolean = false;

  changeName() {
    this.toggle = !this.toggle;
    this.name = this.toggle ? 'Super React' : 'Super Angular'
  }
}

```

Respuestas:

- a) Tanto el nombre del componente Padre como el Hijo se actualizarán.
- b) Solo el nombre del componente Padre se actualizará.
- c) Solo el nombre del componente Hijo se actualizará.
- d) Ninguno se actualizará.

Proyecto 1: Escribir un programa en TypeScript que recorra un arreglo y genere un histograma en base a los números de este. El arreglo se llama myArray y contiene 10 elementos que corresponden a números enteros del 1 al 5. Un histograma representa que tanto un elemento aparece en un conjunto de datos (Debe mostrar la frecuencia para todos los números del 1 al 5, incluso si no están presentes en el arreglo). Por ejemplo, para el arreglo: myArray:=(1,2,1,3,3,1,2,1,5,1) el histograma se vería así:

```

1: *****
2: **
3: **
4:
5: *

```

El código para declarar y poblar myArray ya está ahí, puede editarlo para probar con otros valores y puede hacer clic en el botón de actualizar junto a él para volver al valor original que se utilizará para validar su código durante la prueba. Asegúrese de que los resultados se impriman exactamente como aparecen aquí, ya que incluso un espacio faltante o sobrante puede marcar la pregunta como incorrecta (Notar espacio entre los ":" y el primer asterisco).

Nota: Se debe de enviar el archivo compilable con el proyecto.

Proyecto 2: Escribir un programa utilizando TypeScript que imprima una X construida a base de la letra X y utilizar el carácter de subrayado como espacio. El tamaño de la x se basa en una variable n que indicará el tamaño de la letra para imprimir (en una matriz de n x n). Por ejemplo, para n: = 5 se obtiene:

```
X__X
_X_X_
 _X_
_X_X_
X__X
```

y para n:=6 se obtiene:

```
X___X
_X_X_
 _XX_
 _XX_
_X_X_
X___X
```

Si n es igual a cero imprimir "ERROR"

El código para el tamaño de n ya está ahí, puede editarlo para probar con otros valores y puede hacer clic en el botón de actualización junto a él para volver al valor original que se utiliza para validar su código durante la prueba. Tenga en cuenta que el código debe imprimir los resultados exactamente como se muestra con el fin de que la pregunta sea considerada valida durante la prueba (El caracter "X" en mayúscula y el guion bajo "_" para los espacios).

Nota: Se debe de enviar el archivo compilable con el proyecto.